Agile by the Numbers: What’s Really Going On Out There

Scott W. Ambler
Chief Methodologist for Agile and Lean, IBM Rational
www.ibm.com/developerworks/blogs/page/ambler
twitter.com/scottwambler

© 2010 IBM Corporation

The Surveys
- All survey data, original questions, and summary slide decks can be downloaded from www.ambysoft.com/surveys/
- If you can’t look at the original questions and analyze the data yourself, how can you trust the survey results?
- Some surveys were done via Dr. Dobb’s Journal (DDJ), a community with a wide range of readers, not just Agilists
- Some surveys, the Ambysoft ones, focused on just the agile community
- The source survey for each chart is indicated using graphics such as:
  - DDJ 2009 State of the IT Union
  - Ambysoft 2009 Agile Practices

Warning: You’ll be presented with a lot of information really quickly!

Most Effective Practices: Top 10 (out of 30)

<table>
<thead>
<tr>
<th>Practice</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous Integration</td>
<td>5%</td>
</tr>
<tr>
<td>Daily Stand Up Meeting</td>
<td>47%</td>
</tr>
<tr>
<td>Developer TDD</td>
<td>47%</td>
</tr>
<tr>
<td>Iteration Planning</td>
<td>44%</td>
</tr>
<tr>
<td>Code Refactoring</td>
<td>43%</td>
</tr>
<tr>
<td>Retrospectives</td>
<td>39%</td>
</tr>
<tr>
<td>Pair Programming</td>
<td>36%</td>
</tr>
<tr>
<td>Active Stakeholder Participation</td>
<td>36%</td>
</tr>
<tr>
<td>Potentially Shippable Software</td>
<td>15%</td>
</tr>
<tr>
<td>Burndown Tracking</td>
<td>14%</td>
</tr>
</tbody>
</table>

Ambysoft 2009 Agile Practices
Why agile?

A Few More Thoughts About Success:
- 41% of people believe that canceling a troubled project is a success
- 69% of people had been on a project that they knew would fail right at the very beginning

Agenda
- The Surveys
- Agile Practices
- Adoption Rates
- Success Rates
  - Management
    - Governance
    - Development and Quality
    - Modeling and Documentation
    - Communication
    - Scaling agile
    - Parting Thoughts

An organization’s typical approach to initial estimates on software development projects:

Strategies which project teams use to stay out of trouble or when in trouble to help get them out of it:
- “Questionable” strategies:
  - 19% pad the budget
  - 63% de-scope towards the end of the project to meet deadline
  - 34% ask for extra funds to complete the projects
  - 72% extend the schedule to deliver promised scope
  - 39% avoid scope creep wherever possible via a “change control/management” process
  - 10% change the original estimate to reflect the actuals
  - 18% change the original schedule to reflect the actuals
- Ethical strategies:
  - 12% take a “stage gate” approach to funding
  - 13% have a flexible budget from the beginning of the project
  - 26% have a flexible schedule from the beginning of the project
  - 32% have flexible scope from the beginning of the project

On average, the actual costs of software development projects compared to estimates:

On average, the actuals came in within a +/- 19% range.

© 2010 IBM Corporation

14% of people had been on a project that they knew would fail right at the very beginning.

Within 5% range:
- 12%

Within 10% range:
- 13%

Within 15% range:
- 16%

Within 20% range:
- 16%

Within 25% range:
- 14%

Within 50% range:
- 11%

No initial estimates:
- 21%

DDJ State of the IT Union July 2009

DDJ State of the IT Union July 2009

DDJ State of the IT Union July 2009

© 2010 IBM Corporation

14% of people had been on a project that they knew would fail right at the very beginning.

Within 5% range:
- 12%

Within 10% range:
- 13%

Within 15% range:
- 16%

Within 20% range:
- 16%

Within 25% range:
- 14%

Within 50% range:
- 11%

No initial estimates:
- 21%

DDJ State of the IT Union July 2009

DDJ State of the IT Union July 2009

DDJ State of the IT Union July 2009

© 2010 IBM Corporation
### Approach to Initial Estimation

<table>
<thead>
<tr>
<th>Estimation Type</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>No initial estimate at all</td>
<td>8%</td>
</tr>
<tr>
<td>High-level estimate based on traditional technique</td>
<td>28%</td>
</tr>
<tr>
<td>High-level estimate based on agile estimation</td>
<td>25%</td>
</tr>
<tr>
<td>Detailed estimate based on traditional technique</td>
<td>5%</td>
</tr>
<tr>
<td>Detailed estimate based on agile estimation</td>
<td>7%</td>
</tr>
<tr>
<td>Don’t know</td>
<td>3%</td>
</tr>
<tr>
<td>Detailed estimate based on agile estimation</td>
<td>6%</td>
</tr>
<tr>
<td>Detailed estimate based on traditional technique</td>
<td>5%</td>
</tr>
<tr>
<td>Don’t know</td>
<td>3%</td>
</tr>
</tbody>
</table>

### How long did it take your project team to get started? (Average: 3.9 weeks)

<table>
<thead>
<tr>
<th>Time</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;1 Week</td>
<td>7%</td>
</tr>
<tr>
<td>1 Week</td>
<td>10%</td>
</tr>
<tr>
<td>2 Weeks</td>
<td>12%</td>
</tr>
<tr>
<td>3 Weeks</td>
<td>12%</td>
</tr>
<tr>
<td>4 Weeks</td>
<td>11%</td>
</tr>
<tr>
<td>5-6 Weeks</td>
<td>11%</td>
</tr>
<tr>
<td>7-8 Weeks</td>
<td>11%</td>
</tr>
<tr>
<td>&gt; 8 Weeks</td>
<td>12%</td>
</tr>
<tr>
<td>Don’t Know</td>
<td>10%</td>
</tr>
</tbody>
</table>

### Justifying Agile Projects

81% had to justify their projects in some manner

- Show Stakeholder Concurrence: 48%
- Show Technical Feasibility: 46%
- Considered Commercial Packages: 14%
- Considered Offshoring: 5%
- Estimate ROI: 29%
- Considered Operational Feasibility: 21%
- Estimate NPV: 5%

### Agile Project Management Principles

1. Working software is the primary measure of progress for our agile teams
2. Our agile teams are allowed to work at a sustainable pace
3. Our agile teams identity what done means at the beginning of each iteration
4. Our agile teams only take credit for work that is actually done at the end of each iteration

### Project Management Practices

- Iteration planning (3.54)
- Daily Scrum Meeting (3.29)
- Prioritized worklist (3.08)
- High-level release planning (2.19)
- Retrospectives (1.84)
- One Product Owner (1.55)
- Burndown chart (1.51)
- Potentially Shippable Software (1.51)
- Status Reports (1.15)
- Story Board with Task Breakdowns (0.83)

### Cultural Environment

1. We build agile teams around motivated individuals
2. Our agile teams are provided with the env. and support that they need to succeed
3. Our agile teams are trusted to get the job done
4. Our agile teams are self-organizing
Strongly Agree
Agree
Neutral
Disagree
Strongly Disagree

Ambysoft 2008 Practices and Principles

Process Improvement
1. At regular intervals the team reflects on how to become more effective in future iterations
2. The team actually adjusts its behavior in the next iteration by focusing on the highest priority items

Reflection
Adjustment

Ambysoft 2009 Governance

Agenda
- The Surveys
- Agile Practices
- Adoption Rates
- Success Rates
- Management

Governance
- Development and Quality
- Modeling and Documentation
- Communication
- Scaling agile
- Partnering

Length of Iterations (% respondents)
30% have iterations between 7 and 8 weeks in length

No Iterations
> 8 Weeks
5-6 Weeks
4 Weeks
3 Weeks
2 Weeks
1 Week
< 1 Week

Ambysoft 2008 Agile Adoption

How would you rate your IT governance program?

Too early to tell
Generally helps
Neither helpful nor harmful
Generally harmful
Don’t Know
No IT governance Program

Ambysoft 2009 Governance

The risks and responsibilities (R&R) defined for various groups within your organization?

Defined for Development Teams
Defined for Operations and Support
Defined for Stakeholders
Undefined, but Part of Culture
Undefined, and we Need This
Undefined, Don’t Need This

Do your project teams collect metrics to enable project monitoring by senior management?

No, majority manual
Yes, majority automated
Don’t Know
Agenda
- The Surveys
- Agile Practices
- Adoption Rates
- Success Rates
- Management
- Governance

Development Practices
- Coding Standards (2.30)
- Collective Code Ownership (1.97)
- Continuous integration (1.94)
- Database standards (1.86)
- UI standards (1.65)
- Pair programming (-1.34)

Quality Practices
- Code Refactoring (1.79)
- UI Testing (1.54)
- Automated Developer Testing (1.08)
- TDD (-0.08)
- UI Refactoring (-0.22)
- Database refactoring (-0.31)
- Automated Acceptance Testing (-0.87)
- Database regression testing (-1.03)
- Executable Specs (-1.43)

Value of Various Work Products on Agile Teams (out of 5)
- Working Software (4.22)
- Source Code (4.22)
- Developer Tests (3.96)
- Whiteboard Sketches (3.96)
- Iteration Task List (3.87)
- Customer Tests (3.87)
- Arch Spec - High Level (3.66)
- Defect Reports (3.61)
- Velocity (metric) (3.56)
- Requirements Spec - High Level (3.52)
- Use Cases - Light (3.52)
- Test Plan (3.39)
- Burn Down Chart (3.35)
- Paper models (3.22)
- Use Cases - Detailed (2.96)
- Requirements Spec - Detailed (2.90)
- Arch Spec - Detailed (2.88)
- Gantt Chart - High Level (2.70)
- Gantt Chart - Detailed (2.21)
Agenda

- The Surveys
- Agile Practices
- Adoption Rates
- Success Rates
- Management
- Governance
- Development and Quality

Modeling and Documentation

- Communication
- Scaling agile
- Parting Thoughts

Agile Approach to Initial Requirements

- 76% High-level initial requirements modeling
- 30% Detailed initial requirements modeling
- 12% Have initial requirements models supplied to them
- 15% Use enterprise models as a reference
- 88% Do some sort of initial modeling or have initial models supplied or leverage reference models

Agile Approach to Initial Architecture

- 70% High-level initial architecture modeling
- 25% Detailed initial architecture/design modeling
- 12% Have initial architecture/design models supplied to them
- 14% Use enterprise models as a reference
- 85% Do some sort of initial modeling or have initial models supplied or leverage reference models

Primary Approach to Modeling

<table>
<thead>
<tr>
<th>Method</th>
<th>No Modeling</th>
<th>Sketch to Think and Communicate</th>
<th>Sketch and Capture Key Diagrams</th>
<th>SBMT for Docs</th>
<th>SBMT to Generate Code</th>
<th>SBMT for Full Trip Engineering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agile</td>
<td>🟥</td>
<td>🟥</td>
<td>🟥</td>
<td>🟥</td>
<td>🟥</td>
<td>🟥</td>
</tr>
<tr>
<td>Iterative</td>
<td>🟥</td>
<td>🟥</td>
<td>🟥</td>
<td>🟥</td>
<td>🟥</td>
<td>🟥</td>
</tr>
<tr>
<td>Traditional</td>
<td>🟥</td>
<td>🟥</td>
<td>🟥</td>
<td>🟥</td>
<td>🵜</td>
<td>🟥</td>
</tr>
<tr>
<td>Ad-Hoc</td>
<td>🟥</td>
<td>🟥</td>
<td>🟥</td>
<td>🟥</td>
<td>🵜</td>
<td>🵜</td>
</tr>
</tbody>
</table>

0% 20% 40% 60% 80% 100%

Did you need to produce a vision document (or similar) as part of project initiation?

- No: 37%
- Yes: 56%
- Don't Know: 7%

© 2010 IBM Corporation
1. We do some initial requirements modeling at the beginning of agile projects for scoping and planning purposes.
2. The requirements details emerge over time on our agile projects.
3. Our agile teams have an understanding of the correct balance of documentation or other artifacts for delivery.

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

- Requirements and Documentation

- Design and Architecture

- Modeling vs TDD: Primary Strategy for Requirements Specification Is/Was (%)
**Percentage of Teams Creating Deliverable Documentation**

<table>
<thead>
<tr>
<th>Method</th>
<th>Ad-Hoc</th>
<th>Traditional</th>
<th>Iterative</th>
<th>Agile</th>
</tr>
</thead>
<tbody>
<tr>
<td>User manual</td>
<td>30%</td>
<td>40%</td>
<td>50%</td>
<td>60%</td>
</tr>
<tr>
<td>Training material</td>
<td>20%</td>
<td>30%</td>
<td>40%</td>
<td>50%</td>
</tr>
<tr>
<td>System Overview doc</td>
<td>10%</td>
<td>20%</td>
<td>30%</td>
<td>40%</td>
</tr>
<tr>
<td>Operations doc</td>
<td>0%</td>
<td>10%</td>
<td>20%</td>
<td>30%</td>
</tr>
</tbody>
</table>

**What is the quality of the deliverable documentation produced by a development team?**

Rating: -10 (very low) to 10 (very high)

<table>
<thead>
<tr>
<th>Method</th>
<th>Iterative</th>
<th>Agile</th>
<th>Traditional</th>
<th>Ad-Hoc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feedback Cycle</td>
<td>3.9</td>
<td>-1.3</td>
<td>-1.3</td>
<td>-0.6</td>
</tr>
</tbody>
</table>

And the “quality” of the documentation is the same.

**Face to face (F2F)**

- Iterative: 4.06
- Agile: 4.25
- Traditional: 4.10
- Ad-Hoc: 3.9

**Effectiveness of Communication Strategies**

- Videoconferencing: 1.5
- Overview diagrams: 1.2
- Online chat: 1.1
- Email: 1.0
- Detailed Documentation: 0.5

**Feedback Cycle**

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Our agile project teams welcome new or changing requirements, even just before delivery.
3. Project Stakeholders work closely with our agile teams and are readily available.
4. At regular intervals our agile teams demonstrate potentially shippable software to their stakeholders.

**Agenda**

- The Surveys
- Agile Practices
- Adoption Rates
- Success Rates
- Management
- Strategy
- Development and Quality
- Modeling and Documentation
- Communication
- Scaling agile
- Parting Thoughts

© 2010 IBM Corporation
Does your team have to comply to industry regulations?

Yes 33%
No 60%
Don't Know 7%

Does your team follow a CMMI compliant agile process?

No 78%
Yes 9%
Don't Know 13%

Agile and Legacy Systems

<table>
<thead>
<tr>
<th>Category</th>
<th>Ambyssoft 2009 Agile Project Initiation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Working With Legacy in Some Way</td>
<td>78%</td>
</tr>
<tr>
<td>Integrating With Legacy Systems</td>
<td>57%</td>
</tr>
<tr>
<td>Evolving Legacy Systems</td>
<td>51%</td>
</tr>
<tr>
<td>Working with Legacy Data</td>
<td>45%</td>
</tr>
</tbody>
</table>

Agile scaling factors (of the Agile Scaling Model)

Disciplined Agile Delivery

- Team size
- Compliance requirement
- Geographical distribution
- Domain Complexity
- Enterprise discipline
- Organizational complexity
- Technical complexity

- Under 15 developers
- Over 100 developers
- Critical, cost effective
- Strategic
- Enterprise Scale
- Collaborative
- Homogeneous
- Rigid
- Flexible
Organizational challenges faced when adopting agile

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Waterfall culture</td>
<td>44%</td>
</tr>
<tr>
<td>Stakeholder Involvement</td>
<td>32%</td>
</tr>
<tr>
<td>T&amp;E</td>
<td>33%</td>
</tr>
<tr>
<td>Lack of Trust</td>
<td>32%</td>
</tr>
<tr>
<td>C&amp;C Culture</td>
<td>31%</td>
</tr>
<tr>
<td>Specialization</td>
<td>29%</td>
</tr>
<tr>
<td>Other visions</td>
<td>15%</td>
</tr>
<tr>
<td>Stakeholder Resistance</td>
<td>14%</td>
</tr>
<tr>
<td>Mgmt Resistance</td>
<td>14%</td>
</tr>
</tbody>
</table>

Source: Dr Dobb’s November 2009 State of the IT Union Survey

What do you think about the concept of becoming a "certified master" after taking a two-day course?

- 10% The Certification is Obviously Meaningless
- 12% Respect the Certification
- 12% Don’t Care Either Way
- 78% Everything is Perfect

Source: Dr Dobb’s November 2009 State of the IT Union Survey

Question the Rhetoric

- There appears to be a difference between what people say they are doing and what they are doing.
- Many of the concerns that the traditional community has regarding agile don’t appear to hold true.
- There are many unfounded beliefs in both the traditional and the agile communities.
- In the end, you need to identify what works well for you ➔ Every organization is different.

Thank you

scott_ambler [at] ca.ibm.com
twitter.com/scottwambler
www.ibm.com/rational/agile/
www.jazz.net